**Distributed Continuous Integration**

Nick Draper1, Owen Arnold1, Stuart Campbell2,

Martyn Gigg1,Anders Markvardsen3, Pete Peterson2

1Tessella, 26 The Quadrant, Abingdon, Oxfordshire, UK

2Neutron Data Analysis and Visualization Division, ORNL, USA3ISIS, RAL, STFC, UK

Continuous integration is one of the keys to ensuring that software remains of a high quality without requiring hoards of dedicated testers.

Several years ago, on Mantid[1], we starting using the Jenkins[2] C.I. server as a means of controlling the various builds and tests required for the project. The main server for the project was located at ISIS with a second instance containing subset of the jobs running at SNS. The main problems with this setup were:

* poor/no access to server when ISIS network connection was unstable;
* each instance only had access to its own slave machines to run the builds;
* duplication of hardware at both sites.

Our solution has been to move the Jenkins instance to an externally-hosted server[3]. This provides a single point of control where the network status of one facility no longer impacts on the other. It also allows the C.I. server to share hardware resources from multiple facilities so that build jobs are then distributed over any connected slaves regardless of location, providing more power for additional builds and resilience against network outages at a single facility.

**References**

1. <http://www.mantidproject.org>
2. [http://jenkins-ci.org](http://jenkins-ci.org/)
3. [https://www.linode.com](https://www.linode.com/)

Email corresponding author: nick.draper@stfc.ac.uk Preference: Poster

Key theme: Best Practice in software engineering.